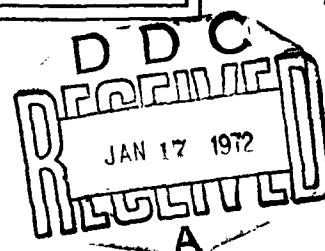# THE INSTITUTE OF STATISTICS

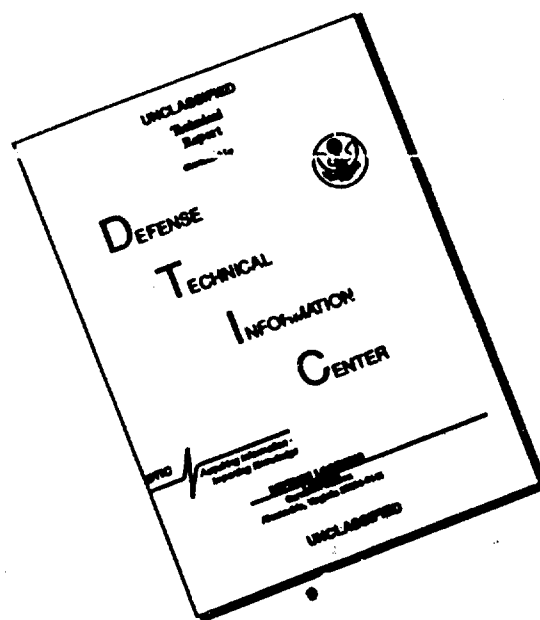## THE CONSOLIDATED UNIVERSITY OF NORTH CAROLINA

DEPARTMENT OF STATISTICS

Chapel Hill, North Carolina

32

DISCLAIMER NOTICE

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Department of Statistics University of North Carolina Chapel Hill, North Carolina 27514	Unclassified
	2b. GROUP

3. REPORT TITLE

Proximate linear programming: An experimental study of a modified simplex algorithm for solving linear programs with inexact data

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

5. AUTHOR(S) (First name, middle initial, last name)

Gould, Floyd J.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November, 1971	-28-	-06-

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-67-A-0321-0003 (NRO 47-096)	Institute of Statistics Mimeo Series Number 789
8b. PROJECT NO.	
	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

The distribution of this report is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Chief of Naval Research Program, Dept. of Navy, 800 N. Quincy Street Arlington, Va. 22217

13. ABSTRACT

A modified simplex method has been developed for attacking large linear programs with inexact data in the right hand sides. Results from performing a limited series of computational experiments are reported. These results indicate reductions in computer time of 30 to 70 per cent over the ordinary simplex method.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Linear Programming						
Large scale systems						
Inexact data						
Tolerances						
Fuzzy right hand sides						
Proximate problem						
Computational experiments						
Applications						

D D C

Proximate Linear Programming:

An Experimental Study of a Modified Simplex Algorithm

For Solving Linear Programs with Inexact Data

F. J. Gould[1]

Department of Statistics
University of North Carolina at Chapel Hill

PROXIMATE LINEAR PROGRAMMING:
AN EXPERIMENTAL STUDY OF A MODIFIED SIMPLEX ALGORITHM
FOR SOLVING LINEAR PROGRAMS WITH INEXACT DATA

F. J. Gould[1]

1. Introduction

In recent years linear programming has become an enormously important and widely used tool for solving a variety of problems encountered in the practice of operations research. The original Simplex method (which we henceforth refer to as the ordinary method) as presented by Dantzig [2] is a robust algorithm which comprises the basic architecture of many commonly used linear programming codes. For other early contributions, see Charnes [1] and Dantzig, Orden and Wolfe [3]. Prominent empirical studies, authored by Wolfe and Cutler [6], and Kuhn and Quandt [4], have compared and reported upon certain variants of the ordinary method. These reported results concern the effects of alternative rules for such features as choice of the pivot element, choice of the form of the inverse, choice of the phase 1 procedure, etc. An advanced exposition of various extensions appears in Orchard-Hays [5].

At the present time, though the ordinary Simplex method is considerably efficient, it is nevertheless true that efforts to solve large problems are typically expensive, and new frontiers in linear programming can be identified with attempts to hasten convergence to optimal solutions of such problems. In

this paper we explore a particular approach to achieving greater efficiency for large scale problems. In particular, a new variant of the ordinary Simplex method is proposed as a possible algorithm for solving a newly defined class of large inexact problems.

As a starting point, we note that in linear programming applications it is not highly unusual for the data of the problem to be inexact. There are many causes for imprecision of data. Some classes of real world problems in fact appear to be impervious to the analyst's most ponderous efforts to cast them into a precise optimization model. Such problems often represent large scale decision models of systems with a highly diffuse and even tentative or speculative structure. The purpose of the model is frequently to enable a decision maker to gain insight for purposes of gross planning, explore tradeoffs, and merely *improve* rather than gain optimality. The very notion of sharp optimization may be a matter of forensics - at least within the limits on the precision of the data. Linear programs of this nature might descriptively be dubbed as "proximate linear programs". In spite of this special quality of imprecision in many of our models, and in spite of its inevitable accentuation with the growing interest in social and urban problems, the prepotency of the engineering disciplines has continued to almost completely shape the mode of inquiry in rigorous operations research. Looking at a special class of large scale imprecise problems - linear programs with tolerances in some of the data - it is the objective of this study to attempt to develop an algorithm which might intelligently recognize and exploit the special structure afforded by inexact data. In short, it was hoped that an algorithm could be developed which would tend to be more efficient than the ordinary Simplex method when the problem data are inexact, and which would reduce to the ordinary method when the data are precise.

To define the goal more explicitly, assume that the right hand sides of the given problem are known only within certain user specified tolerances. Let ϵ be a vector and let the implied right hand side windows be denoted by $\bar{b} \pm \epsilon$. All other problem data are assumed to be exact. Our goal, then, is to develop an efficient algorithm for solving the *proximate problem*: find a nonnegative x^* such that, for some b in the rectangle $[\bar{b}-\epsilon, \bar{b}+\epsilon]$,

(i) $Ax^* \leq b$, and

(ii) $c^T x^* \geq c^T x$ for every nonnegative x such that $Ax \leq b$.

It has been our experience that optimization theorists tend to have difficulty in catching the unfamiliar flavor of this problem. Analysts have an irresistable tendency to be greedy, and the typically encountered question is, "Why not use the right hand side $\bar{b} + \epsilon$ and get the maximum possible return?" The point here is that in the ground rules for the model under consideration the right hand side is not a policy variable under the analyst's control. It is an exogenous "fuzzy" parameter, and the value $\bar{b} + \epsilon$ is no more worthy of consideration than any other value in the rectangle $[\bar{b}-\epsilon, \bar{b}+\epsilon]$. Consequently, we propose an effort which capitalizes upon ignorance. We wish to solve the linear program with the assumption of some underlying $b \in [\bar{b}-\epsilon, \bar{b}+\epsilon]$, with indifference as to which one, with an algorithm which hopefully can economize on consumption of computer time.[1] The algorithm proposed in this paper is heavy handed in the sense that it relies upon well known ideas. It is a modification of the ordinary Simplex method which reduces to the latter when all

---

[1] There may be other immediate applications for such an algorithm as opposed to direct attack of an imprecise problem. For example, suppose the ultimate objective of a study is to do a sensitivity analysis over a range of right hand sides. The proposed algorithm could generate a fast optimal solution for some right hand side in the range. Post optimality analyses can then be done. See the appendix for other possible applications.

tolerances are zero (i.e., when the right hand side data are exact). It is not possible at this writing to state whether or not the objective of increased efficiency has been realized with any generality over many classes of programs. Initial computational results indicate

(i) the method appears to differ insignificantly (in terms of computation time) from the ordinary Simplex method for A matrices of size 25×25, with tolerances up to $\pm 30\%$.

(ii) With A matrices of size 100×100, 200×150 and 100×200, when the Simplex method is applied to $Ax \leq \bar{b}$, and when the modified method is applied to $Ax \leq b$, $\bar{b}-\epsilon \leq b \leq \bar{b}+\epsilon$, there are obvious differences in computation time. With tolerances (ϵ) of roughly 30% the modified method appears to reach optimality in 30 to 70 per cent of the amount of time taken by the ordinary method, the realized reductions being dependent upon the sparsity and the percentage of negative coefficients in A. In this case, objective values between Simplex optimality and modified optimality tend to differ relatively by average amounts of 2 to 10 per cent.

These results and others are presented in detail in the final section of the paper. It will be obvious that the modified method requires, on the average, far fewer pivots than the ordinary Simplex algorithm. However, each pivot requires more work. It tentatively appears that performance of the method improves with problem size, is better for positive A matrices, and is not helped by sparseness. It should be stated that while our computational experience is at present far too limited to draw any conclusions, there would seem to be adequate justification for further empirical inquiry into the usefulness of the modified method, expecially as regards the performance on large problems. Also, theoretic possibilities exist for allowing tolerances in the other data of the problem, for improving phase 1, for further improving or modifying the proposed technique for tolerances in the right hand sides

(e.g., building in a procedure for equality constraints), and for applying the proposed technique to other contexts. Several such applications are discussed in the appendix to this paper.

It may be helpful at the outset to make a rough descriptive comparison between the method to be advanced and the ordinary Simplex method. Consider the polytope given by $Ax \leq b_1$, $x \geq 0$, and the larger containing polytope given by $Ax \leq b_2$, $x \geq 0$. These two polytopes define the proximate problem of interest. In our context, given a mean for the right hand side interval, say $\bar{b}$, and given a vector of tolerances, say ϵ, we shall have $b_1 = \bar{b} - \epsilon \leq \bar{b} \leq \bar{b} + \epsilon = b_2$. We begin with a phase 1, if necessary, on the polytope $Ax \leq b_1$, $x \geq 0$, in order to find an initial extreme point, say E_0. Beginning at E_0, the ordinary Simplex method would change basis by exchanging 2 columns. The column to be entered distinguishes an edge. The column to be removed is chosen in such a way that, speaking geometrically, there is a motion along the distinguished edge from E_0 to an adjacent extreme point E_1. In the modified method one will also change basis by exchanging 2 columns. The column to be entered is chosen as in the ordinary Simplex method, thus distinguishing the same edge. (A criterion other than the ordinary entry criterion could be used.) However, the remove column is not necessarily chosen so as to step to an adjacent extreme point. Rather, it is chosen according to the criterion: Take as large a step (along the distinguished edge) as possible to a new basic solution with the following properties

   (1) the activities x_j, $j = 1,\ldots,n$ must remain nonnegative

   (ii) the new solution may be infeasible with respect to any subset of the right hand sides b_{1i}, but not with respect to any of the right hand sides b_{2i}; that is, the new solution may violate some of the constraints with right hand side b_{1i}, and hence lie outside the polytope $Ax \leq b_1$, $x \geq 0$, but it must not be outside the polytope $Ax \leq b_2$, $x \geq 0$.

(In general, such a basic solution will lie past the extreme point adjacent to E_0. Consequently, we achieve greater increase in the objective function.) At this new basic solution, the violated b_{1i} right hand sides are translated to b_{2i} values. We then are at an extreme point, say $\hat{E}$, of the polytope

$$\sum_{j=1}^{m} a_{ij} x_j \leq b_{1i}, \qquad i \in N_1$$

$$\sum_{j=1}^{m} a_{ij} x_j \leq b_{2i}, \qquad i \in N_2$$

$$x \geq 0$$

where N_2 indexes the constraints which have been assigned the new right hand side value b_{2i}. Beginning, then, from $\hat{E}$, the same method is repeated to obtain a new point $\hat{\hat{E}}$ which will be an extreme point of some (possibly new) polytope. At each extreme point the sets N_1 and N_2 are updated as required. Initially, at point E_0, in the case where all constraints have tolerances, the set N_1 indexes all the constraints and N_2 is empty. (The procedure, however, will allow some of the constraints to be exact. If they are all exact, the method reduces to the ordinary Simplex method.) As the algorithm repeats, the indices tend to transfer from the set N_1 to the set N_2. At some stages the sets may remain the same. Computational experience indicates that large transfers from N_1 to N_2 tend to occur early in the game as opposed to later. Also, as expected, larger tolerances encourage larger transfers and better performance for the modified method. Figure 2 provides a sketch of the geometry of the new method.

2. Notation

Let A be an $m \times n$ matrix with columns $P_1, P_2, \ldots, P_n$. Let $\bar{b}$ be in R^m, c in R^n, and let ε denote a given nonnegative m-vector of tolerances (for example, the components of ε may be given by $\varepsilon_i = .20\bar{b}_i$). The problem to be solved is

P: find nonnegative $x^* \in R^n$ such that for some $b \in [\bar{b}-\varepsilon, \bar{b}+\varepsilon]$

(i) $Ax^* \leq b$, and

(ii) $c^T x^* \geq c^T x$ for every $x \geq 0$ such that $Ax \leq b$.

Any such x^* will be called an optimal solution to P.

Let $X = \{1,2,\ldots,n\}$ denote the indices of the activities and let $S = \{n+1,\ldots,n+m\}$ be the set of slack indices. Let the fuzzy slacks be indexed by $F = \{i \in S: \varepsilon_{i-n} > 0\}$, and let the exact slacks be indexed by $E = S-F$. Let $b_1 = \bar{b} - \varepsilon$, $b_2 = \bar{b} + \varepsilon$, and let β be the nonnegative m-vector given by $\beta = b_2 - b_1 = 2\varepsilon$.

Let $\hat{A}$ denote the A matrix augmented in the usual way with an $m \times m$ identity matrix and denote the last m columns of $\hat{A}$ as $P_{n+1},\ldots,P_{n+m}$. A basis, say B, will correspond to m columns of $\hat{A}$, say $P_{I_1}, P_{I_2}, \ldots, P_{I_m}$. Corresponding to each basis B will be a point in R^n with basic coordinates $x_B = (x_{I_1}, x_{I_2}, \ldots, x_{I_m})$. Nonbasic coordinates are always assigned the value zero. Also, corresponding to each basis, B, is a tableau with rows indexed from 1 to $m+1$ and columns indexed from 0 to $m+n$. The entries of the zero column of the tableau corresponding to B are defined by $x_{j0} = x_{I_j}$, $j = 1,\ldots,m$; $x_{m+1,0} = c_B^T x_B$. The entries in the remaining columns, excluding the last row, are given by the coefficients x_{ij} in the expressions $P_j = \sum_{i=1}^{m} x_{ij} P_{I_i}$, $j = 1,\ldots,n+m$. The last row of the tableau, excluding $x_{m+1,0}$, consists of the usual reduced costs.

In the method to be described, as in the ordinary Simplex method, a basis B will be replaced with a basis $\hat{B}$ by a simple exchange. That is, some vector P_{I_r} (corresponding to the rth row) is replaced by a vector P_e. Thus, the new basis $\hat{B}$ is given by $P_{\hat{I}_1}, P_{\hat{I}_2}, \ldots, P_{\hat{I}_m}$, where $\hat{I}_r = e$ and $\hat{I}_j = I_j$ for $j \neq r$. It will be convenient to let I denote the set $\{I_1, \ldots, I_m\}$ of indices of a current basis B, and to let R_k, for $k \in I$, denote the row of the current tableau such that $x_{R_k 0} = x_k$. Thus, we have the notational relations

$$R_{I_j} = j \quad (\text{i.e.,} \quad x_{I_j} = x_{j0}), \quad j \in \{1, \ldots, m\}$$

$$I_{R_j} = j \quad (\text{i.e.,} \quad x_{R_j 0} = x_j), \quad j \in I.$$

Finally, the sets J_1 and J_2 will denote "current partitions" of the set S (i.e., $J_1 \cup J_2 = S$, $J_1 \cap J_2 = \phi$). The sense of "current partition" will become clear in the sequel, for we shall be interested in problems of the form

$$(1) \quad \begin{cases} \max c^T x, \quad \text{subject to} \\ \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_{1i}, \quad n + i \in J_1 \\ \sum_{j=1}^{n} a_{ij} x_j + x_{n-i} = b_{2i}, \quad n + i \in J_2 \\ x \geq 0. \end{cases}$$

3. The Algorithm

The modified method, flow charted in Figure 1, can be described in 9 steps.

1. *Initial Step.* Set $J_2 = E$, $J_1 = F$. Do a phase 1 on (1) to obtain an initial feasible basis B and corresponding basic feasible solution x. GO TO 2.

2. If all reduced costs are nonpositive, TERMINATE. Otherwise, choose an enter column, e, as in the ordinary Simplex method, to be one which has a maximum reduced cost. Define θ_1 and T_1 as follows:

$$\theta_1 = \min_{j \in I} [x_j/x_{R_j e} : x_{R_j e} > 0] = x_t/x_{R_t e}, \quad t \in T_1.$$

That is, T_1 denotes the set of basic indices which tie for the determination of θ_1.

GO TO 3.

3. If $T_1 \subseteq J_1$, set $i = 2$ and GO TO 4. Otherwise, select an index $I_r \in T_1$, obtain a new basis $\hat{B}$ by replacing P_{I_r} with P_e, and perform a normal pivot operation on the tableau, including an ordinary transformation of the values x_{j0}, $j = 1, \ldots, m$.

Then GO TO 2.

4. Define θ_i and T_i as follows:

$$\theta_i = \min_{j \in I - \bigcup_{p=1}^{i-1} T_p} [x_j/x_{R_j e} : x_{R_j e} > 0] = x_t/x_{R_t e}, \quad t \in T_i.$$

That is, T_i denotes the set of basic indices which tie for the determination of θ_i. If $T_i \cap J_2 \cap F = \phi$ then GO TO 5. Otherwise GO TO 9.

5. If $\theta_i \leq \min[(x_j + \beta_{j-n})/x_{R_j e} : j \in \bigcup_{p=1}^{i-1} T_p]$, then let $i = i + 1$ and GO TO 6. Otherwise,

GO TO 9.

6. If $T_{i-1} \cap E = \phi$, GO TO 7. Otherwise,

GO TO 9.

7. If $T_{i-1} \cap X = \phi$, GO TO 8. Otherwise,

GO TO 9.

8. If $\bigcup_{j=1}^{i-1} T_j \neq I$, GO TO 4. Otherwise,

GO TO 9.

9. Let $\hat{\theta} = \theta_{i-1} = x_{I_r}/x_{re} = x_{r0}/x_{re}$ for some $I_r \in T_{i-1}$. Replace P_{I_r} with P_e to obtain a new basis. The set of new basic indices is $I - \{I_r\} \cup \{e\}$. Obtain new values of the new basic variables from the expressions

$$(2) \quad \hat{x}_j = \begin{cases} x_j - \hat{\theta}x_{R_j e}, & j \in I - \bigcup_{p=1}^{i-2} T_p \\ x_j - \hat{\theta}x_{R_j e} + \beta_{j-n}, & j \in \bigcup_{p=1}^{i-2} T_p \\ \hat{\theta}, & j = e. \end{cases}$$

As always, the nonbasic $\hat{x}_j$ are set equal to zero. The new tableau values x_{j0}, $j = 1,\ldots,m$ are given by $x_{j0} = \hat{x}_{I_j}$ as determined from (2). Transform the remainder of the tableau by an ordinary pivot operation. Update the sets J_1 and J_2 according to

$$J_1 = J_1 - \bigcup_{p=1}^{i-2} T_p$$

$$J_2 = J_2 \cup (\bigcup_{p=1}^{i-2} T_p).$$

GO TO 2.

Remarks on the Algorithm:

(i) Note that in step 5 the subscript $j - n$ is positive since $(U_{p=1}^{i-1} T_p) \cap X = \phi$. This equality holds because the only way to enter step 5 is from step 4 and the only way to enter step 4 is from step 3 or step 8. The route $3 \rightarrow 4 \rightarrow 5$ implies $i = 2$ and $T_1 \subseteq J_1$. Hence, $T_1 \cap X = \phi$. The route $8 \rightarrow 4 \rightarrow 5$ implies $i \geq 3$ and, from step 7, $T_j \cap X = \phi$ for each j such that $2 \leq j \leq i - 1$.

(ii) By convention, for $i < 3$, the set $U_{p=1}^{i-2} T_p$ is empty. In step 9, for $i \geq 3$, $U_{p=1}^{i-2} T_p \subseteq J_1$, but T_{i-1} need not be a subset of J_1. The latter assertion follows from steps 6 and 7. The first assertion is argued as follows. Suppose step 9 is entered with $i = i_0$. Then the tests in steps 4, 6 and 7 must have been passed for $i = i_0 - 2$.
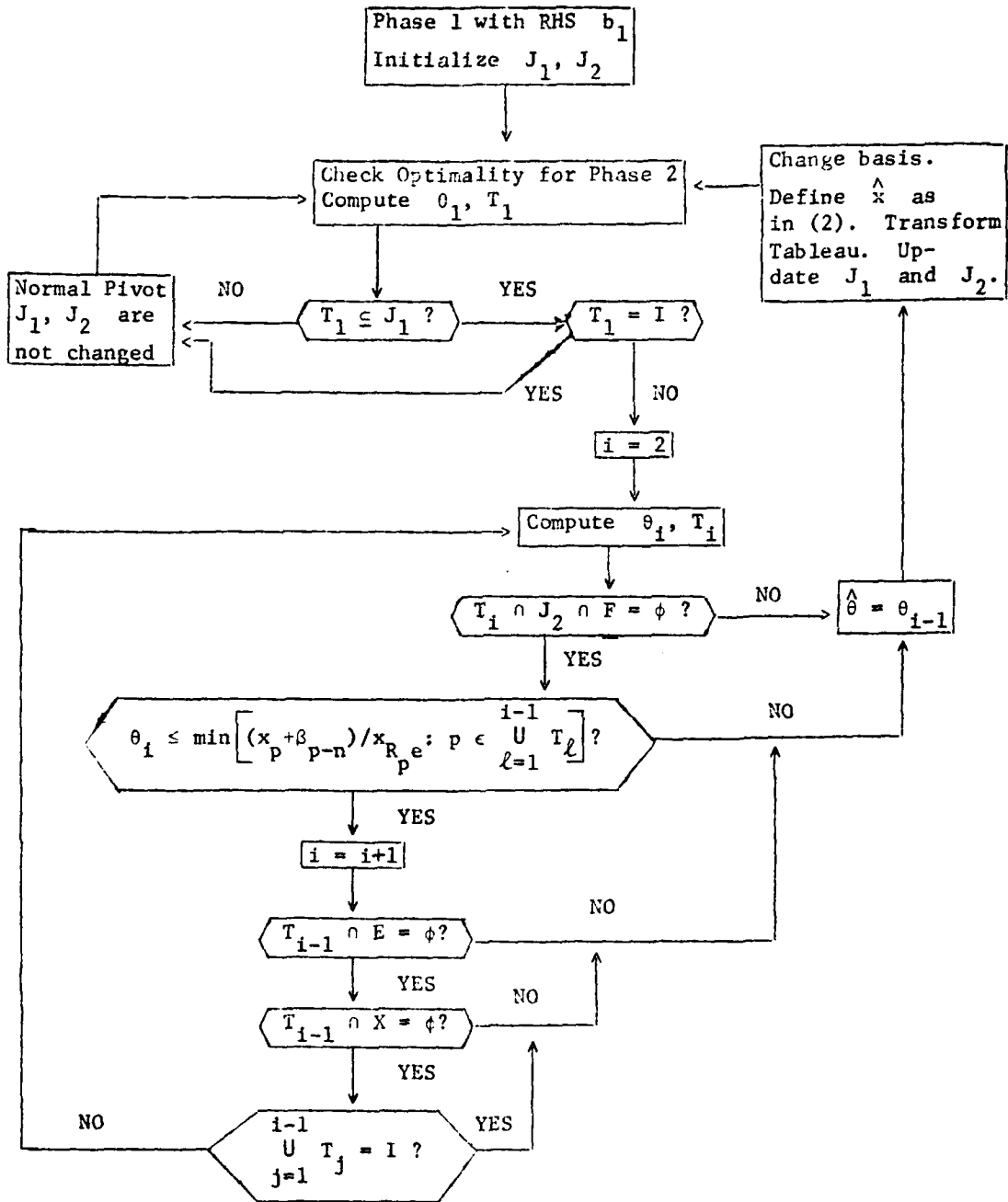
Figure 1: Flow Chart for Modified Algorithm

4. Optimality Proof

Theorem: After each pivot operation on the tableau, as prescribed either in step 3 or in step 9, the current entires x_{j0}, $j = 1,\ldots,m$, along with the current values of J_1 and J_2, represent a basic feasible solution to (1).

Proof: It is clear that either in step 3 or in step 9 the basis B is replaced with a new basis $\hat{B}$. The old basic indices I are replaced with the indices $\hat{I} = I - \{I_r\} \cup \{e\}$. We now show that the current entries x_{j0} are feasible, $j = 1,\ldots,m$. By definition, this requires a proof that $\hat{x}_j \geq 0$, $j \in \hat{I}$. Since the pivot operation in step 3 is an ordinary Simplex maneuver, it suffices to consider the values $\hat{x}_j$ as prescribed in (2). From step 4, replacing i with $i - 1$,

$$0 \leq \hat{\theta} = \theta_{i-1} = \min_{x_{R_j e} > 0} \left[x_j / x_{R_j e} : j \in I - \bigcup_{p=1}^{i-2} T_p \right]$$

From step 5, replacing i with $i - 1$,

$$\hat{\theta} = \theta_{i-1} \leq \min \left[(x_j + \beta_{j-n}) / x_{R_j e} : j \in \bigcup_{p=1}^{i-2} T_p \right].$$

Thus

$$x_j - \hat{\theta} x_{R_j e} \geq 0, \qquad j \in I - \bigcup_{p=1}^{i-2} T_p$$

$$x_j - \hat{\theta} x_{R_j e} + \beta_{j-n} \geq 0, \qquad j \in \bigcup_{p=1}^{i-2} T_p.$$

and $\hat{\theta} \geq 0$. Consequently $\hat{x}$ is nonnegative. Finally, given the current values of J_1 and J_2, we show that $\hat{x}$ is a solution to (1) and hence, by the above remarks, a basic feasible solution. We have,

$$\sum_{j=1}^{m+n} \hat{x}_j P_j = \sum_{\substack{j \in I - \bigcup\limits_{p=1}^{i-2} T_p}} (x_j - \hat{\theta} x_{R_j e}) P_j + \sum_{\substack{j \in \bigcup\limits_{p=1}^{i-2} T_p}} (x_j - \hat{\theta} x_{R_j e} + \beta_{j-n}) P_j + \hat{\theta} P_e$$

$$= \sum_{j \in I} x_j P_j - \hat{\theta} \sum_{j \in I} x_{R_j e} P_j + \hat{\theta} P_e$$

$$+ \sum_{\substack{k \in \bigcup\limits_{p=1}^{i-2} T_p}} \beta_{k-n} P_k = \sum_{j \in I} x_j P_j + \sum_{\substack{k \in \bigcup\limits_{p=1}^{i-2} T_p}} \beta_{k-n} P_k.$$

Now let

$$b = \sum_{j \in I} x_j P_j, \qquad \tilde{b} = \sum_{\substack{k \in \bigcup\limits_{p=1}^{i-2} T_p}} \beta_{k-n} P_k.$$

Hence

$$b_j = \begin{cases} b_{1j}, & n + j \in J_1 \\ \\ b_{2j}, & n + j \in J_2 \end{cases}$$

where J_1 and J_2 are the values of the previous partition. In the expression for $\tilde{b}$ let $n + j = k$. Hence

$$\tilde{b} = \sum_{\substack{n+j \in \bigcup\limits_{p=1}^{i-2} T_p}} \beta_j P_{n+j},$$

and

$$\tilde{b}_j = \begin{cases} \beta_j, & n + j \in \bigcup\limits_{p=1}^{i-2} T_p \\ \\ 0, & \text{if not.} \end{cases}$$

We have therefore deduced that

$\sum_{j=1}^{m+n} \hat{x}_j P_j = \hat{b}$, where $\hat{b} = b + \tilde{b}$. That is,

$$\hat{b}_j = \begin{cases} b_{1j} + \beta_j = b_{2j}, & n + j \in J_1 \cap \left(\bigcup_{p=1}^{i-2} T_p \right) \\ b_{1j}, & n + j \in J_1 - \bigcup_{p=1}^{i-2} T_p \\ b_{2j}, & n + j \in J_2. \end{cases}$$

From the remarks following the algorithm, $\bigcup_{p=1}^{i-2} T_p \subseteq J_1$ and therefore $\hat{b}$ simplifies to

$$\hat{b}_j = \begin{cases} b_{1j}, & n + j \in J_1 - \bigcup_{p=1}^{i-2} T_p \\ b_{2j}, & n + j \in J_2 \cup \left(\bigcup_{p=1}^{i-2} T_p \right). \end{cases} \qquad \square$$

Corollary: The algorithm terminates in an optimal solution to the problem P.

Proof: Each tableau represents a basic feasible solution to some problem of the form (1). Termination can only occur when all reduced costs are non-negative. From ordinary Simplex theory, this is an optimal tableau for the problem of form (1) which is currently represented. This implies that we have reached an optimal solution to problem P. $\qquad \square$

5. An Example

In Figure 2, a problem with 13 inequality constraints in 2 variables is illustrated. The solid lines represent the b_1 and b_2 hulls. Each plane is labeled with the index of the corresponding slack variable. At the initial point, A, we have

$$X = \{1,2\}$$

$$S = \{3,4,\ldots\ldots,15\}$$

$$E = \phi$$

$$F = S$$

$$J_1 = S$$

$$J_2 = \phi$$

The basic indices, I, are given by $\{i \notin \{3,4\}\}$

$$e = 3$$

$$T_1 = \{5\}, \quad T_2 = \{6\}, \quad T_3 = \{7\}, \quad T_4 = \{8\}, \quad T_5 = \{9\}$$

$$I_r = 8 \in T_4 = T_{i-1}; \quad i = 5; \quad \hat{\theta} = \theta_4.$$

If x_j, $j = 1,\ldots,15$ represents the point A, then point B is given by the coordinates $\hat{x}_j$, $j = 1,\ldots,15$, where

$$\hat{x}_j = x_j - \hat{\theta} x_{R_j e}, \quad j \in I - \{5,6,7\}$$

$$\hat{x}_j = x_j - \hat{\theta} x_{R_j e} + \beta_{j-2}, \quad j \in \{5,6,7\}$$

$$\hat{x}_3 = \hat{\theta}.$$

At point B, the nonbasic components $\hat{x}_4$ and $\hat{x}_8$ each have the value zero. Also at point B, we have

$$J_1 = S - \{5,6,7\}$$

$$J_2 = \{5,6,7\}.$$

Thus, point B is a basic feasible solution to the problem

$$\max c_1 x_1 + c_2 x_2, \quad \text{subject to}$$

$$\sum_{j=1}^{2} a_{ij} x_j + x_{n+i} = b_{1i}, \quad i \in S - \{5,6,7\}$$

$$\sum_{j=1}^{2} a_{ij} x_j + x_{n+i} = b_{2i}, \quad i = 5,6,7$$

$$x_j \geq 0, \quad j = 1,2,\ldots,15.$$

From point B the modified method takes us to point C, with $e = 4$,
$I_r = 12$, $i = 5$, $\hat{\theta} = \theta_4$. It is seen that point C is an optimal solution
to the proximate problem. It can be seen that point C is an exact solution,
i.e., an optimal basic feasible solution, to the problem

$$\max c_1 x_1 + c_2 x_2, \quad \text{subject to}$$

$$\sum_{j=1}^{2} a_{ij} x_j + x_{n+i} = b_{1i}, \quad i = 3, 4, 8, 12, 13, 14, 15.$$

This hull is sketched in Figure 3.
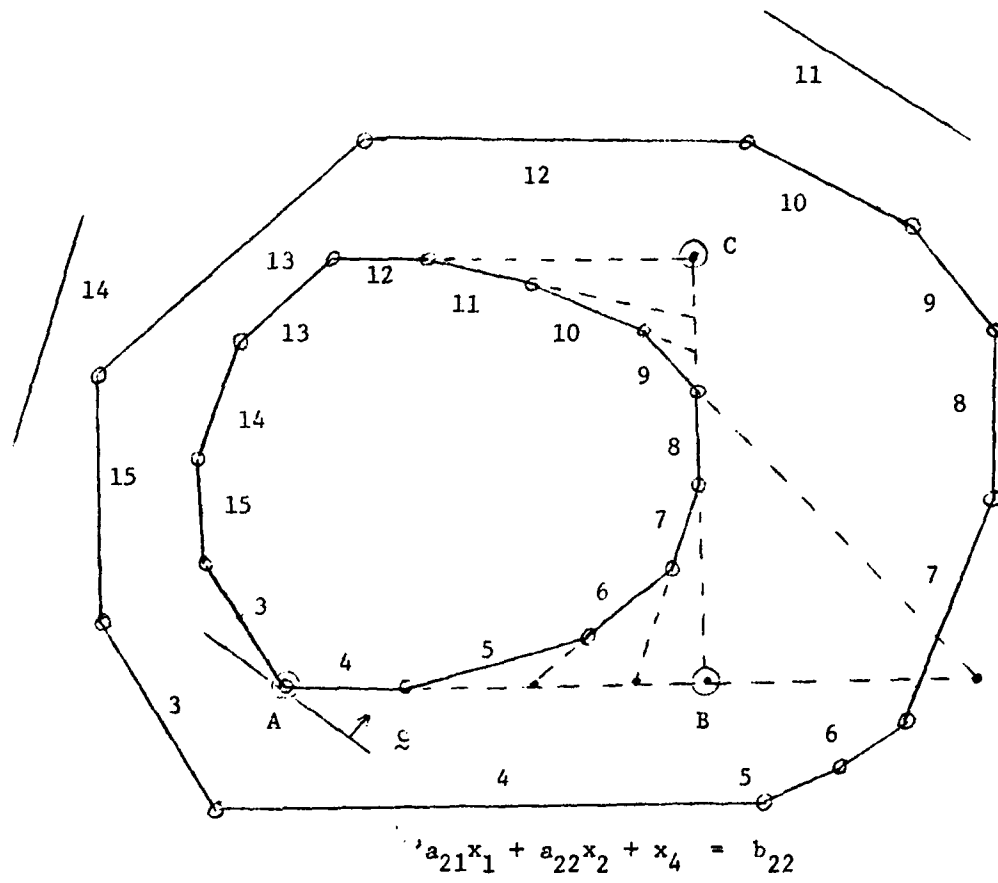
$$'a_{21}x_1 + a_{22}x_2 + x_4 = b_{22}$$

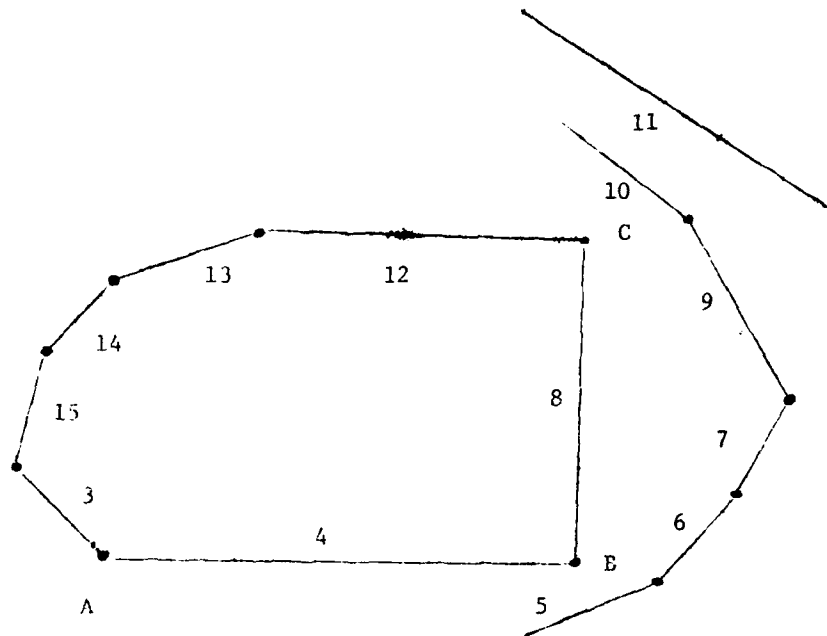FIGURE 2: Modified Method Pivots from A to B to C

FIGURE 3:

The Solutions Space Corresponding to the Final Values of J_1, J_2

6. Numerical Experiments

In this section, we report the outcome of comparisons between the modified method and the ordinary method on 131 problems. All runs were performed on the IBM Model 370/165 Computer. The experiments involved randomly generated data with A matrices of four sizes: 25×25, 100×100, 200×150, and 100×200. Table I summarizes the input data, where D is a random number between 0 and 1. In all cases the objective function coefficients were given by $c = 15 + 10D$. The right hand side for the ordinary Simplex runs is denoted by the vector $\bar{b}$, and the ordinary runs are designated in Table I by $\varepsilon = 0$. These runs were performed on an in-house all-in-core full tableau code employing the usual maximum reduced cost criterion for the choice of enter vector. As a validity check on the in-house code, identical problems have been run on MPS, the IBM linear programming package, with identical results to at least 5 decimal places. The modified Simplex algorithm, as elaborated in Section 3, was programmed as a special option of the in-house code. By the choice of test problems, as seen in Table I, an initial feasible basis of full slack was available for all runs. It was therefore necessary to consider only phase 2 and all reported results are based on starts from the initial full slack basis. The test problem characteristics were chosen in such a way at to explore the effects of

    (i)   the size of A

    (ii)   the dispersion of the a_{ij}'s

    (iii) the percentage of negative a_{ij}'s

    (iv)   the percentage of zeros in A.

In Table I, P_z denotes the percentage of zeroes in A. These were randomly assigned to entries in A and remaining entries were then drawn from the appropriate interval. For each problem type, as defined by a row of Table I,

5 randomly generated problems were run, with the exception of the 200×150 case. For each of these latter problems only 3 random tests were conducted. The results are presented in Table II, where

M_T = mean computation time in seconds

M_I = mean number of iterations

R = mean computation time for modified Simplex divided by mean computation time for ordinary Simplex.

ERROR = the mean value of $\dfrac{100|OBS - OBM|}{OBS}$, where

OBS = optimal objective value obtained with ordinary Simplex (right hand side $\bar{b}$)

OBM = optimal objective value obtained with modified Simplex (given the appropriate tolerance about $\bar{b}$).

TABLE I: TEST DATA

Run	Dim. of A	a_{ij}	$\bar{b}$	ϵ	Pz(%)	b_1	b_2
1	25 × 25	.01 + 1600D	23,490	0	0	----	----
2	"	"	23,053	.15 $\bar{b}$	0	20,000	26,511
3	"	"	23,490	.31 $\bar{b}$	0	16,200	30,780
4	100 × 100	.01 + 500D	23,490	0	0	----	----
5	"	"	23,053	.15 $\bar{b}$	0	20,000	26,511
6	"	"	23,490	.31 $\bar{b}$	0	16,200	30,780
7	"	"	23,912	.43 $\bar{b}$	0	13,630	34,194
8	"	"	23,490	.80 $\bar{b}$	0	4,698	42,282
9	100 × 100	.01 + 500D	23,490	0	25	----	----
10	"	"	"	.31 $\bar{b}$	"	16,200	30,780
11	100 × 100	1 + 50D	23,490	0	0	----	----
12	"	"	"	.31 $\bar{b}$	"	16,200	30,780
13	100 × 100	1 + 50D	23,490	0	25	----	----
14	"	"	"	.31 $\bar{b}$	"	16,200	30,780
15	100 × 100	.01 + 2000D	23,490	0	0	----	----
16	"	"	"	.31 $\bar{b}$	"	16,200	30,780
17	"	"	"	0	25	----	----
18	"	"	"	.31 $\bar{b}$	"	16,200	30,780
19	100 × 100	-125 + 500D	23,490	0	0	----	----
20	"	"	"	.31 $\bar{b}$	0	16,200	30,780
21	100 × 100	-125 + 500D	23,490	0	25	----	----
22	"	"	"	.31 $\bar{b}$	"	16,200	30,780
23	100 × 100	-250 + 1000D	23,490	0	0	----	----
24	"	"	"	.31 $\bar{b}$	"	16,200	30,780
25	200 × 150	1 + 598D	57,971+ 14,493D	0	0	----	----
26	"	"	"	.31 $\bar{b}$	0	40,000+ 10,000D	75,492+ 18,986D
27	100 × 200	.01 + 500D	23,490	0	0	----	----
28	"	"	"	.31 $\bar{b}$	"	16,200	30,780

TABLE II: TEST RESULTS

Run	ε	M_T	M_I	R	ERROR (%)
1	0	2.5	25	----	---
2	.15 $\bar{b}$	2.7	18	1.08	3.5
3	.31 $\bar{b}$	2.5	10	1	7
4	0	63.2	350	----	---
5	.15 $\bar{b}$	41.2	149	.64	1.9
6	.31 $\bar{b}$	25.6	57	.40	3
7	.43 $\bar{b}$	20.3	32	.32	9
8	.80 $\bar{b}$	15.0	4.4	.24	8.1
9	0	45.5	239	----	---
10	.31 $\bar{b}$	25.4	73	.55	4.9
11	0	59	333	----	---
12	.31 $\bar{b}$	19.9	47	.33	4.4
13	0	48.5	242	----	---
14	.31 $\bar{b}$	26.8	77	.55	5.4
15	0	59.0	346	----	---
16	.31 $\bar{b}$	23.0	50	.39	2.8
17	0	47.7	240	----	---
18	.31 $\bar{b}$	23.6	75	.49	5.0
19	0	58.5	321	----	---
20	.31 $\bar{b}$	38.5	153	.65	2.7
21	0	48.7	252	----	---
22	.31 $\bar{b}$	34.5	95	.64	5.3
23	0	59.6	335	----	---
24	.31 $\bar{b}$	39.1	152	.65	2.8
25	0	204	394	----	---
26	.31 $\bar{b}$	83.5	75	.40	2.9
27	0	151	470	----	---
28	.31 $\bar{b}$	54	69	.35	3.2

The most meaningful comparison between the modified method and the ordinary Simplex method is in terms of computation time. On this basis, with respect to a *ceteris paribus* analysis, the results in Table II suggest the following observations:

1. Increasing the tolerance appears to considerably increase the performance of the modified method, without introducing great relative error in the optimal objective values (runs 4, 5, 6, 7, 9).

2. For the small 25×25 problems, up to a tolerance of $\pm 30\%$, the ordinary method was superior to the modified method.

3. For all 100×100, 200×150 and 100×200 runs, the modified method was superior.

4. For 100×100 matrices, the ordinary Simplex performance was independent of dispersion on the a_{ij}'s and of the percentage of negative a_{ij}'s (runs 4, 11, 15, 19, 23).

5. For positive matrices (such as runs 6, 11, 15), the modified method gave its best performance, with a suggestion that tighter dispersion on the a_{ij}'s improves its efficiency. There is a strong suggestion (runs 20 and 24) that the presence of negative a_{ij}'s damps the efficiency of the modified method and therefore, in view of observation 4 above, decreases its advantage over the ordinary method. It should be noted that from the geometry of the modified method it might be intuitively expected to perform best on a Bayes hull.

6. The addition of zeros to the A matrix uniformly improved the ordinary Simplex performance (compare the M_T values in 4 with 9; 11 with 13; 15 with 17; 19 with 21).

7. The addition of zeros to the A matrix does not appear to influence the efficiency of the modified method (compare the M_T values in 6 with 10; 12 with 14; 16 with 18, and 20 with 22), but, because of observation 6

above, the zeros diminish the advantage of the modified method over the ordinary method.

8.   Runs 21 and 22, which in this size problem give the poorest relative advantage to the modified method, tend to support observations 5 and 7.   In runs 21 and 22, the Λ matrix contained 25% zeros and about 18% negative coefficients. With the zeros enhancing the ordinary Simplex performance and the negativities dulling the modified performance the relative advantage (the R value) is only .7.

Appendix

There are two possibly interesting special uses for the modified method which we here briefly describe.

1. Suppose it is desired to obtain a dual feasible tableau for the problem max $c^T x$, subject to $Ax \leq \bar{b}$, $x \geq 0$. That is, we seek a basis B such that, setting $\omega^T = c_B^T B^{-1}$, we have $\omega^T A \geq c^T$ and $\omega \geq 0$. One can obtain such a basis B by selecting an especially large tolerance (the geometric distance between the inner and outer hull is the key factor) and using the modified method to obtain fast convergence to some optimal solution to some problem (e.g., run 8 in Tables I and II). The final basis will yield feasible dual variables and the final tableau will be dual feasible for any right hand side. If it were then desired to use the dual method one would begin with the first column given by $B^{-1} \bar{b}$.

2. Assume for convenience that all constraints are normalized and suppose it is desired to obtain a near optimal solution to the problem max $c^T x$, subject to $Ax \leq b_2$, $x \geq 0$, within a prespecified objective value accuracy. In particular, suppose we seek a feasible solution, say x_0, such that $Ax_0 \leq b_2$, $x_0 \geq 0$, and $c^T x_0 \leq c^T \hat{x} \leq c^T x_0 + \alpha$ for some specified value α, where $\hat{x}$ is a true solution to max $c^T x$, $Ax \leq b_2$, $x \geq 0$. Select a large tolerance ϵ and let $\bar{b} = b_2 - \epsilon$, $b_1 = \bar{b} - \epsilon$. Use the modified method to obtain fast convergence to an optimal solution to some proximate problem, say x^*, with feasible basis B_*. If $B_*^{-1} b_2 \geq 0$, then the basic components of $\hat{x}$ are given by $B_*^{-1} b_2$. Otherwise, one could proceed with the dual algorithm, possibly modified for inexact data.

Acknowledgment

References

1. Charnes, A., "Lectures on Linear Programming" (mimeo), USDAF – Carnegie Institute of Technology Research Project on Intra-Firm Planning and Behavior (Pittsburgh: Carnegie Institute of Technology, GSIA, 1950, 1951, 1952).

2. Dantzig, G. B., "Maximization of a Linear Function of Variables Subject to Linear Inequalities," *Activity Analysis of Production and Allocation*, T. C. Koopmans, ed., Wiley (1951) 339-347.

3. Dantzig, Orden, and Wolfe, "The Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Restraints," *Pacific Journal of Mathematics*, 5 (1955) 183-195.

4. Kuhn, H. and Quandt, R., " An Experimental Study of the Simplex Method," *Proceedings of Symposia in Applied Math.*, 15 (1963) American Math. Society, 107-124.

5. Orchard-Hays, W., *Advanced Linear Programming Computing Techniques*, McGraw-Hill, (1968).

6. Wolfe, P. and Cutler, L., "Experiments in Linear Programming," *Recent Advances in Mathematical Programming*, R. L. Graves and Philip Wolfe, eds., McGraw-Hill (1963) 177-200.